Section C.2

2025-06-04

Table of contents


```
######################################################################################
# discrim() is a customized R function to determine discriminant functions.
# By submitting the following lines in R, the function will be defined.
# PLEASE DO NOT BE CONCERNED ABOUT HOW THIS FUNCTION IS DEFINED.
discrim <- function(Y, group){
Y <- data.matrix(Y)
group <- as.factor(group)
m1 <- manova(Y ~ group)
nu.h <- summary(m1)$stats[1]
nu.e <- summary(m1)$stats[2]
p <- ncol(Y)
SS <- summary(m1)$SS
E.inv.H <- solve(SS$Residuals) %*% SS$group
eig <- eigen(E.inv.H)
s <- min(nu.h, p)
lambda <- Re(eig$values[1:s])
a <- Re(eig$vectors[,1:s])
a.star <- (sqrt(diag(SS$Residuals/nu.e)) * a)
return(list("a"=a, "a.stand"=a.star))
}

######################################################################################
# discr.sig() is a customized R function to test significance of discriminant functions.
# By submitting the following lines in R, the function will be defined.
# PLEASE DO NOT BE CONCERNED ABOUT HOW THIS FUNCTION IS DEFINED.
discr.sig <- function(Y, group){
```

```
Y <- data.matrix(Y)
group <- as.factor(group)
m1 <- manova(Y ~ group)
sums <- summary(m1)
evals <- sums$Eigenvalues
nu.e <- m1$df
nu.h <- m1$rank-1
k <- nu.h + 1
p <- ncol(m1$coef)
N <- nu.e + nu.h + 1
s <- min(p, nu.h)
lam <- numeric(s)
dfs <- numeric(s)
for(m in 1:s){
lam[m] <- prod(1/(1+evals[m:s]))
dfs[m] <- (p-m+1)*(k-m)
}
V <- -(N - 1 - .5*(p+k))*log(lam)
p.val <- 1 - pchisq(V, dfs)
out <- cbind(Lambda=lam, V, p.values=p.val)
dimnames(out)[[1]] <- paste("LD",1:s,sep="")
return(out)
}

##############################################################################################
# partial.f() is a customized R function to test for significance of additional variables.
# By submitting the following lines in R, the function will be defined.
# PLEASE DO NOT BE CONCERNED ABOUT HOW THIS FUNCTION IS DEFINED.
partial.F <- function(Y, group){
Y <- data.matrix(Y)
group <- as.factor(group)
p <- ncol(Y)
m1 <- manova(Y ~ group)
nu.e <- m1$df
nu.h <- m1$rank-1
Lambda.p <- summary(m1,test="Wilks")$stats[3]
Lambda.p1 <- numeric(p)
for(i in 1:p){
dat <- data.matrix(Y[,-i])
m2 <- manova(dat ~ group)
Lambda.p1[i] <- summary(m2,test="Wilks")$stats[3]
}
```

```
Lambda <- Lambda.p / Lambda.p1
F.stat <- ((1 - Lambda) / Lambda) * ((nu.e - p + 1)/nu.h)
p.val <- 1 - pf(F.stat, nu.h, nu.e - p + 1)
out <- cbind(Lambda, F.stat, p.value = p.val)
dimnames(out)[[1]] <- dimnames(Y)[[2]]
ord <- rev(order(out[,2]))
return(out[ord,])
}

################################################################################################
# discr.plot() is a customized R function to visualize discriminant functions.
# By submitting the following lines in R, the function will be defined.
# PLEASE DO NOT BE CONCERNED ABOUT HOW THIS FUNCTION IS DEFINED.
discr.plot <- function(Y, group, leg = NULL){
a <- discrim(Y, group)$a
z <- data.matrix(Y) %*% a
plot(z[,1], z[,2], type = "n", xlab = "LD1", ylab="LD2")
for(i in 1:length(unique(group))){
points(z[group == unique(group)[i],1],
z[group == unique(group)[i],2], pch = i)
}
if(is.null(leg)) leg <- as.character(unique(group))
legend("topright",legend = leg,pch=1:length(unique(group)))
}
#################################################################
lin.class <- function(Y,group){
# Install MASS package if not already installed
if (!require("MASS")) install.packages("MASS")
library(MASS)
Y <- data.matrix(Y)
group <- as.factor(group)
p <- ncol(Y)
m1 <- manova(Y ~ group)
nu.e <- m1$df
nu.h <- m1$rank-1
Sp <- summary(m1)$SS$Residual/(nu.e)
cio <- 1:m1$rank
c.mat <- matrix(nrow=m1$rank,ncol=p,0)
for (i in 1:m1$rank) {
cio[i] <- -.5*t(lda(Y,group)$means[i,])%*%solve(Sp)%*%
lda(Y,group)$means[i,]
c.mat[i,] <- t(lda(Y,group)$means[i,])%*%solve(Sp)
```

```
}
return(list("coefs"=c.mat,"c.0"=cio))
}
#################################################################
rates <- function(data,group,method="l") {
if (!require("MASS")) install.packages("MASS")
library(MASS)
data <- as.matrix(data)
group <- as.matrix(group)
da.obj <- lda(data,group)
if (method=="q") {
da.obj <- qda(data,group)
method <- "QDA"
}
tab <- table(original=group,predicted=predict(da.obj)$class)
if (method=="l") method <- "LDA"
cor.rate <- sum(predict(da.obj)$class==group)/nrow(data)
er.rate <- 1-cor.rate
return(list("Correct Class Rate"=cor.rate,"Error Rate"=er.rate,
"Method"=method,"Confusion Matrix"=tab))
}
```

C.2: Discriminant Analysis

Discriminant Functions and Variable Importance

```
nhanes <- read.csv("NHANES3_419.csv")
nhanes$SBPRANK <- as.factor(nhanes$SBPRANK)

# Omit height variable
nhanes <- nhanes[, c("SBPRANK", "HSAGEIR", "BMPWTLBS", "PEPMNK5R", "TCP")]

X <- nhanes[, -1]
y <- nhanes[, 1]

discrim(X, y)$a.stand
```

```
           [,1]      [,2]
[1,] -10.784324 -8.677779
[2,]  -3.772346  7.113163
```

```
[3,]  -7.754878  8.769770
[4,]   2.881487  2.341885
```

Let:

- y_1 = HSAGEIR (Age) (variable 2)
- y_2 = BMPWTLBS (Body Weight) (variable 3)
- y_3 = PEPMNK5R (Average Diastolic BP) (variable 5)
- y_4 = TCP (Serum Cholesterol) (variable 6)

Then the standardized discriminant functions are:

$$LD_1(y) = -10.784y_1 - 3.772y_2 - 7.755y_3 + 2.881y_4$$

$$LD_2(y) = -8.678y_1 + 7.113y_2 + 8.770y_3 + 2.342y_4$$

We now rank the standardized coefficients of each discriminant function, LD_1 and LD_2, by observing their absolute values.

For the first discriminant function (LD_1), y_1 (age) and y_3 (average diastolic BP) are the most important for separating the groups, followed by y_4 (cholesterol) and y_2 (body weight): $y_1 \to y_3 \to y_4 \to y_2$.

For the second discriminant function (LD_2), y_3, y_1, and y_2 are the most important, followed by y_4: $y_3 \to y_1 \to y_2 \to y_4$.

Significance Tests for Discriminant Functions

```r
discr.sig(X, y)
```

```
       Lambda          V   p.values
LD1 0.4856068 147.721810 0.00000000
LD2 0.9596942   8.413261 0.03820007
```

Hypotheses:

- 1st Test (for LD_1):
  - H_0: $\alpha_1 = \alpha_2 = 0$
  - H_a: At least one $\alpha_i \neq 0$
- 2nd Test (for LD_2):

- H_0: $\alpha_2 = 0$
- H_a: $\alpha_2 \neq 0$

Conclusions:

- blabla..

Significance Tests for Additional Variables

```
partial.F(X, y)
```

```
             Lambda    F.stat      p.value
HSAGEIR  0.6046790 66.357652 0.000000e+00
PEPMNK5R 0.7659335 31.018029 1.759481e-12
BMPWTLBS 0.9344024  7.125580 1.021397e-03
TCP      0.9679940  3.356017 3.681948e-02
```

Visualizing Discriminant Functions

```
discr.plot(X, y)
```